Application of Hyperbolic Embedding in Overlay Network Construction

Fragkiskos Papadopoulos
CAIDA
frag@caida.org

D. Krioukov (CAIDA), A. Vahdat (UCSD)
Overview

• Motivation

• A Network Model that Grows in a Hyperbolic Space

• Application to Peer-to-Peer Overlays

• Conclusion
What is Peer-to-Peer (P2P)?

• A model of communication where every node in the network acts alike

• As opposed to the Client-Server model, where one node provides services and other nodes use the services
Example

Underlying network (i.e. router graph)

P2P/Overlay network is formed by the blue links
Advantages of P2P Networks

• Scalability
  – Every peer acts both as a Client and a Server => as demand increases, so does system capacity => scalable
  – Traditional Client-Server sharing: performance deteriorates as the number of clients increases

• No single point of failure
  – Data replication over multiple peers
  – Peers can find data without relying on centralized index servers
Types of P2P Networks

A. Unstructured (e.g. Gnutella, FastTrack)
   - Overlay links are established arbitrarily when a new node joins => simple, no topology maintenance costs
   - To find content: use controlled flooding of queries, random walk variations, etc. => not scalable and no guarantee that peer having content is found

B. Structured or DHTs (e.g. Chord, Kademlia)
   - More structured pattern of overlay links => nodes need to maintain up-to-date information for a set of other nodes
   - Queries are answered with very high probability
A Closer Look at Structured P2P Networks

• Main idea:

(i) Nodes are assigned unique identifiers (ids), e.g. via consistent hashing (e.g. SHA-1 on node IP address)

(ii) Data elements are also assigned unique identifiers using the same function, and are related to the node with the “closest” id

(iii) To find/store content: forward towards the node with the closest id

• Performance example: Consider a network of N peers

Chord requires: $O(\log N)$ routing information, $O(\log N)$ hops, $O(\log^2 N)$ messages per node arrival/departure

➢ Main Problem: How to provide good performance in high churn rates
This Talk

• Discusses the possibility of constructing overlay networks in Hyperbolic Spaces (are there any benefits?)

• To beat existing architectures, we need:

  - Support for any churn rate with minimal information exchange
  - Minimal routing information at nodes
  - Locate content with ~100% success
  - Shortest paths towards the peer responsible for content
Overview

• Motivation
• A Network Model that Grows in a Hyperbolic Space
• Application to Peer-to-Peer Overlays
• Future Work
Hyperbolic Geometry: The Poincaré Disc Model

Poincaré disc Model

Tessellation by triangles
Constructing Scale-free Networks in the Poincaré Disc

- **Perform the following operations (D. Krioukov, F. Papadopoulos, M. Boguna, A. Vahdat, 2008):**

  - Fix the disc radius to $R$ according to $N = \kappa e^{R/2}$ where $N$ is the number of nodes and $\kappa$ used to tune the average degree to a target value.
  - Assign to each node an angular coordinate $\theta$ uniformly distributed in $[0, 2\pi]$
  - Assign to each node a radial coordinate $r$ in $[0, R]$, with probability $\rho(r) = \alpha e^{\alpha r}/(e^{\alpha R} - 1)$
  - Connect every pair of nodes whenever the hyperbolic distance between them is smaller than $R$

Resulting graph is scale-free (power-law degree distribution and strong clustering) with exponent $\gamma = 2\alpha + 1$ ($1/2 \leq \alpha \leq 1$)
Navigation in the Poincaré Disc

- Has been shown (D. Krioukov, F. Papadopoulos, M. Boguna, A. Vahdat, 2008):
  - Greedy routing (i.e. forward packet to the neighbor closer to destination in the hyperbolic space) has >99.9% success probability and stretch $\approx 1$ if $\gamma$ is small.
  - Above still hold in dynamic conditions (random node/link removals), *without the need for updates*

- Question: Can we construct overlay networks in the Poincaré disc?
  - Cannot use the above model as is (assumes network size does not grow)
  - We need a *growing model*
A Growing Model

- Initially there are 0 nodes in the network
- A new arriving node i needs to know:
  
a. The current number of nodes in the network $N(i)$
  
b. A pre-specified (constant) node density value $\delta$, which dictates how the average node degree evolves
  
c. The parameter of the node density distribution $\alpha$, which determines the exponent of the degree distribution
- To connect to the network, node $i$ performs the following operations:

  a. Selects an angular coordinate $\theta$ uniformly distributed in $[0, 2\pi]$
  b. Computes the current disc radius $R(i) = \frac{1}{\alpha} \text{acosh}(1 + \alpha N(i) / 2\pi \delta)$
  c. Selects a radial coordinate $r$ in $[0, R(i)]$, with probability $\rho(r) = \frac{\alpha e^{\alpha r}}{(e^{\alpha R(i)} - 1)}$
  d. Connects to all nodes in the network for which their hyperbolic distance to it is smaller than $R(i)$

- Differences from the static model

  (i) The disc grows $R(i) \sim \ln N(i)$
  (ii) Average degree grows slowly $\sim \ln N(i)$
  (iii) Maximum degree also grows $\sim N(i)^{1/(\gamma-1)}$
Degree Distribution

\[ \delta = 0.003, \ \alpha = 0.6, \ \gamma \approx 2.2 \]

Average degree in the range 3.9-11 as N changes from 100 to 5000
Clustering

![Graph showing clustering with different N values]
Greedy Routing Success Probability

$p_s$ vs. $N$

- $p_s$ increases rapidly initially, then levels off as $N$ increases.
Greedy Routing Av. Number of Hops
Greedy Routing Av. Stretch

Average stretch

N
Overview

• Motivation

• A Network Model that Grows in a Hyperbolic Space

• Application to Peer-to-Peer Overlays

• Future Work
Constructing the Overlay

- **Node id**: hyperbolic coordinates (unique since continuous domain)

- Recall: Arriving node i needs to discover N(i) (to compute R(i)), and its neighbors

- Use “discovery packet”
  - Forward to highest degree neighbor
  - Neighbor writes its own and neighbors’ coordinates
  - Forwards to highest degree neighbor
  - Node that sees discovery packet twice sends it back to node i

- **Power-law graph** = > *the procedure terminates in very few hops*
Constructing the Overlay (Cont.)

- Neighbor discovery also very efficient
- Note, $R(i) \sim \ln N(i) \Rightarrow$ very small error if estimated $N(i)$ is not 100% accurate
Data Operations (in progress)

- **Data element’s id**: also hyperbolic coordinates

- **Store operation**: at the node whose id is closest to the data id (in hyperbolic distance); *How many ids/copies per data?*

- **Search and Retrieve**: perform greedy routing with the data id as the destination
  
  ➢ *Above ensure minimal routing information and shortest path routing (search), better than existing architectures*
  
  ➢ *But, what about success ratio?*

  Success ratio depends on data id(s) and number of copies

  *“Conjecture”*: We need a small number of copies to achieve 100% success ratio
Data Operations (Cont.)

- Churn rate?

  - Nodes leaving the system can assign responsibility for their data to their neighbors

  - “Better nodes” for an item arriving to the system?
Conclusion

• Have presented a network model that grows in a hyperbolic space

• Have demonstrated that the network is scale-free

• Have demonstrated that greedy routing performs exceptionally well as the network grows

• Have discussed the possibility of constructing P2P overlays using this model

• It may be possible that these overlays will outperform all existing architectures, but be aware of the “catch” (power-law degree distribution)

THANK YOU!